

Problema do Troco

Um dos mais clássicos problemas de ciência da computação é o problema do troco em moedas. O problema em si existe desde que os seres humanos começaram a fazer trocas voluntárias, e na era da computação, esse problema parece cada vez mais secundário, mas sua resolução ainda tem muito sentido para questões de aprendizado.

A questão foi primeiramente observada no campo da economia, e a medida que a computação foi tomando conta das atividades rotineiras e repetitivas da vida humana, esse problema foi trazido para ser contemplado no campo da ciência da computação.

O problema de Frobenius, ou simplesmente problema do troco (no inglês, coin change problem) consiste em encontrar uma combinação com o menor número de moedas cuja soma seja igual a uma quantia determinada, partindo da premissa de que há uma lista de moedas válidas que possuem disponibilidade infinita. Levando em conta as denominações padrões da moeda brasileira, o Real, há moedas para R\$0,01, R\$0,05, R\$0,10, R\$0,25, R\$0,50 e R\$1,00. Na prática, o objetivo que leva a resolução do problema é, dada uma certa quantia (geralmente trabalha-se com um número inteiro de centavos), encontrar um caminho de instruções no algoritmo que fornece a quantia com uma quantidade de moedas válidas mínima. Em termos de programação, esta quantidade de moedas geralmente é tida como infinita. Isso faz com que, no programa, para qualquer quantia de centavos necessários para o troco, há uma solução possível, independente do caixa real.

É importante notar que o problema apresentado é levemente similar a outro problema clássico da computação; o problema da mochila, em que se busca colocar produtos que maximizam o valor de uma mochila com uma restrição, geralmente peso ou espaço. De certa forma, o problema do troco é uma inversão do problema da mochila: com um valor fixo para a mochila, é necessário encontrar a combinação de produtos com peso menor que fornece esse valor.

Programação dinâmica

Uma das abordagens mais comuns utilizadas para resolver o problema se dá através dos algoritmos ditos “gulosos”. Este tipo de algoritmo ira buscar apenas a moeda que mais se aproxima do resultado, independente do número de moedas utilizadas na solução. Isso faz com que, em algumas circunstâncias, o método guloso não seja o método de resolução que apresente a solução ótima, ou seja, não é o algoritmo que utiliza a menor quantidade de moedas para chegar ao resultado.

Uma solução que chega mais próxima da requerida pela descrição do problema é a utilização da programação dinâmica, que busca o resultado ótimo de forma diferente, otimizando a recompensa, conhecida na programação também como “função objetivo”, para alcançar uma “visão geral” da solução, em vez de buscar apenas o ganho imediato.

A solução proposta para o problema, utilizando programação dinâmica, envolve desenvolver um algoritmo que, a partir de uma lista de k moedas válidas, encontre qual moeda pode ser escolhida para se encontrar o caminho ótimo que leva até o valor do troco, e repetir esse processo até que o valor total das moedas escolhidas seja o valor total do troco.

Referências

<http://prorum.com/?qa=3250/problema-troco-resolve-abordagem-natural-sempre-funciona>

<https://forum.scriptbrasil.com.br/topic/177133-problema-do-troco/>

https://www.wikiwand.com/pt/Problema_do_troco_de_Frobenius